A Mobile World of Security - the Model

Christine Neuberg (EPFL), Panos Papadimitratos (KTH), Christina Fragouli (EPFL) and Ruediger Urbanke (EPFL)
christine.neuberg@epfl.ch, papadim@kth.se, christina.fragouli@epfl.ch, rudiger.urbanke@epfl.ch

***Abstract*—We propose a novel approach to establish cryptographic keys among mobile users and a networking infrastructure. Our approach comes at a low cost and can either be used as an alternative to existing solutions or can be employed in a complementary way. Our basic observation is that users are often very mobile. As they interact with the infrastructure, each of them leaves a unique trace behind, known both to the users and the infrastructure. We can leverage this shared information to create shared secret keys, with little or no change of existing mobile communication systems. We show that we can achieve (almost passively) a rate of roughly 0.1 bits per second.**

I. Introduction

A core requirement for future wireless systems is to support communication for large numbers of mobile users, while offering security and privacy. As we move from traditional cellular telephony to more fluid and mobile settings, security and privacy become increasingly important and difficult to achieve, especially for highly mobile users.

A basic question is how a roaming device can establish cryptographic keys in a highly volatile mobile environment. A standard yet organizationally complex solution (especially for large-scale multi-domain dynamic systems), a public key infrastructure (PKI), could address this problem. Present-day cellular data systems rely on pre-established secrets (SIM cards) and cross-organizational verification of identity and authorization (e.g., for billing).

Several alternatively schemes, surveyed in Sec. VI, leverage the wireless channel variability, so that two nodes in range of each other utilize reciprocal observation of signals to establish a shared secret key.

Our work is close in spirit, yet we propose a fundamentally different approach: to leverage mobility and its inherent randomness as the basis for key generation. Our starting point is that users are often quite mobile and they interact with networking infrastructures; in doing so, each of them leaves behind a unique trace. Many works seek to identify structure in the mobility of a population, to anticipate user mobility and offer services. Instead, we exploit the inherent randomness of each individual mobility trace. For example, the sequence of base stations a driver's smart phone connects to and the times of these encounters constitute shared information between the user and the infrastructure.

This information allows establishing a shared secret, and renewing it over time thanks to the user mobility. Having such a source of secret keys opens a new range of opportunities for users to address security and privacy needs, complementing existing techniques. In this paper, we investigate the feasibility of this novel approach and we shed light on what rates of secrecy can be achieved. Our main contributions are:

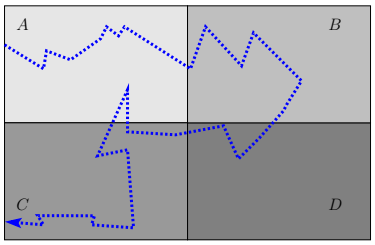

Figure 1. Bob's (the mobile's) itinerary in the area of Alice (the infrastructure), which operates the four base-stations, A,B,C, and D.

1) A new approach for secret key generation, applicable to a wide range of systems that involve mobile nodes and a wireless communication infrastructure. We investigate design choices and propose a backward-compatible protocol with low overhead cost for existing cellular systems.
2) The performance evaluation of our scheme, with analytical modeling. In the analysis, we show that on the one hand, the scheme can be modeled as a renewal process. On the other hand, we develop a relation between different sampling intervals that allows for more efficient evaluation. In addition to pure simulation, we also use a mobility model, that allows for efficient evaluation of a wide range of scenarios. We find that over a sufficiently shot for practical applications acceptable period of time (comparable to average commute times for example) a mobile node and an entity on the back-end of the infrastructure can derive common information, sufficient for establishing a strong secret shared key; even in the presence of a strong adversary.

In the remainder of the paper, Sec. II describes the main idea behind our approach and it introduces basic definitions, Sec. III presents our scheme, Sec. IV introduces our analytical model and Sec. V presents the scheme evaluation. Sec. VI surveys related work before we conclude. In this paper, give a sketch of the ideas and procedures. For more details, we refer the reader to the extended version of the paper [1].

II. Main Idea and Setup

Currently, mobility is perceived primarily as a hurdle to overcome in mobile computing systems that need to support secrecy and secure communication. Our position is that, on the contrary, mobility can be a source of secrecy and thus enable security enhancements. Moreover, such enhancements can be achieved using simple schemes, building on well-established existing infrastructure. We illustrate our approach assuming a cellular infrastructure.

978-1-4244-9848-2/11$26.00©2011 IEEE

A. Main Idea

Consider a user who enters a geographical area at some time t_0 and drives around until he exits the area at a later time t_e. His itinerary can be described by his geographical position at each time t within the interval of interest, $t_0 \leq t \leq t_e$, captured by a two-dimensional curve $L(t)$, as depicted in Fig. 1. Consider this geographical area covered by a cellular infrastructure, where base stations are labeled. During his itinerary, the user connects to various base stations. We can think of the history of connections as a stochastic process $X(t)$. Assume that we sample this process every δ_s seconds. This creates the discrete signal $X[i] \triangleq X(i\delta_s)$. We then keep n samples $X[i]$ in a vector $X \triangleq [X[0], X[1], \ldots, X[n-1]]$. This vector is what our protocol uses as a shared secret between the user and the infrastructure.

B. Adversary

In our model, the adversary is a passive[1] eavesdropper. Its presence, e.g. through the deployment of eavesdropping devices, is assumed bounded, that is, within a part of the entire area. Assuming an arbitrary non-unique label assignement to all base stations, we consider an adversary able to intercept all communication between the user, the infrastructure and all base stations with the same label, where we assume the base stations to be labeled randomly with 4 or more different labels. This defines a particularly strong adversary because, as we discuss in Secs. IV, V, the adversary can intercept all messages with an arbitrary precision to 25% of all base stations. We emphasize that the interception of all messages constitutes a worst-case scenario and the coverage of 25% of all base stations implies vast resources for the adversary.

C. Performance Metrics

In order to evaluate the quality of the generated key, we need to quantify the inherent randomness in the collected vectors X. If the base stations are assigned labels from a set $\mathcal{M}$ of size m, then we can consider this vector X to be a discrete random variable that takes m^n values.

Our performance metric is the *secrecy rate* R_s, that quantifies the number of generated random bits per second. In the presence of an adversary, this quantifies the number of bits, the adversary has no information about. For traces of length n, R_s is calculated as $\frac{\mathbb{H}(X)}{n\delta_s}$, where $\mathbb{H}(X)$ is the entropy of the vectors X. As n increases, this quantity approaches $\frac{\mathcal{H}(X)}{\delta_s}$, where $\mathcal{H}(X)$ is the entropy rate. Recall that the entropy rate is a lower bound of the entropy and takes into account all dependencies, even those that go beyond the boundary of the vector [2]. The entropy rate is for our evaluation the most fundamental measure. Similarly, in the presence of the adversary A, we use the conditional entropies $\frac{\mathbb{H}(X|A)}{n\delta_s}$ and $\frac{\mathcal{H}(X|A)}{\delta_s}$.

[1] An active adversary is orthogonal to our investigation. Clearly, injection of arbitrary messages, e.g., by impersonation of a base station, would introduce errors in our scheme; however, such errors can be detected leading to aborting the key generation. Moreover, cellular networks or commercial Wi-Fi infrastructures are well protected and centrally managed; even though WiFi access point impersonation is perhaps easy, this is much harder for base stations in cellular systems (used as an example system here), which significantly raises the bar for an adversary.

III. Our Scheme

Our scheme establishes a secret key between two entities:

(i) The *infrastructure-side* entity, C, any machine (e.g., server) that lies on the back side of the wireless infrastructure (i.e., on the wire-line network). The wireless infrastructure consists of a set of *base stations*, $\{I\}$, distributed across a geographical area, connected with wire-line links and nodes, such as routers and switches. C can access the $\{I\}$ infrastructure.

(ii) A *mobile node*, V, that roams in the geographical area covered by $\{I\}$. V can connect to base stations in $\{I\}$ when in range, possibly making use of a semi-reliable data link protocol to mitigate communication errors. Over time, V connects to multiple base stations, one at a time.

Each base station is assigned a label from a set $\mathcal{M}$. The *base station labeling* can be conveyed by S to V. Multiple base stations may have the same label.

Trace Collection: To generate a shared key, V and C need to *collect a trace* of data on the connectivity of V with base stations in $\{I\}$. The collected samples $X[i] = X(i\delta_s)$ are stored in what we term the *trace* vector. $X = [X[0],\ X[1], \ldots X[n-1]]$, where $n = \frac{T}{\delta_s}$ and $T = t_e - t_0$ is the total sampling duration. The samples are retained with appropriate logging at C and V.

Trace Consolidation: V and C need to *consolidate* their traces, i.e., account for any errors in the trace creation, so that the key extraction is done on the same trace. To reveal minimal information about X_V and X_S during consolidation, even in the presence of an adversary that perfectly intercepts the communication, well-known mechanisms for privacy amplification can be used [3].

Key Extraction: Finally, S and V use the common information, X_S (which is identical to X_V), to derive a shared secret key $K_{V,S}$. Well known methods as these proposed in [4], [5] can be used.

Design Parameters: Three basic parameters control the trace collection:

1) The *alphabet size*, i.e., the size of the set $\mathcal{M}$, or the number of distinct labels we assign to the base stations.
2) The *sampling interval* δ_s, i.e., how often we sample.
3) The *sampling duration* $T = t_e - t_0$, i.e., for how long we sample.

Parameter choice guidelines: (1) We propose to employ $\mathcal{M}$ of a size equal to the maximum number of neighbors a base station can have; and a labeling that assigns a different label to each neighbor. (2) We suggest selecting a sampling interval of the same order of magnitude as the average sojourn time, e.g. $\frac{1}{4}$ of the average sojourn time. (3) Finally, the sampling duration T can be selected sufficiently large to allow extracting the required number of random bits.

A detailed justification of parameter choices and an example of infrastructure compatible deployment can be found in [1].

978-1-4244-9848-2/11$26.00©2011 IEEE

IV. Modeling and Analysis

We introduce a model to analyze the scheme performance, i.e., the quality of the produced key in terms of entropy and entropy rate (see Sec. II). We use as input only the distribution of sojourn times, which can be collected experimentally. We validate our model in Sec. V, showing that the analytic performance predictions very well agree with our exhaustive simulation results. Thus, our modeling can significantly simplify the performance analysis of the scheme (i.e., the required simulations). More important, it helps understand better what the important parameters for the key generation rate are.

A. Model

Recall that $X(t)$ is the stochastic process describing the labels of the base station that a user is connected to at time t, and $X = [X[0],\ X[1], \dots X[n-1]]$ is the trace of length n, and δ_s is the sampling interval. The process $X(t)$ is quite complicated, and so is X. We therefore introduce a model which is analytically tractable but preserves the key characteristics of $X(t)$.

Let S be the random variable that describes the sojourn time. For a fixed sampling period δ_s, we measure the sojourn time in multiples of δ_s, i.e., $S \in \mathbb{N}\,\delta_s$. E.g., $S = 3\delta_s$ means that for 3 consecutive sampling times a user is connected to a particular base station, but that at the 4-th sampling time instance he has switched. Our model takes as input the distribution of S for a given parameter δ_s; this distribution can be calculated through simulations or modeling.

Definition 1 (Interarrival Times U): As mentioned above, S takes values in $\mathbb{N}\,\delta_s$. Associate to S the integer-valued random variable U, where $p_i = P(U = i) = P(S = i\delta_s),\ i \geq 1$. We call U the *inter-arrival* time.

We now postulate that we can model the connection process by a renewal process with inter-arrival times distributed according to U. With respect to the labels, we assume that for every new connection the label is chosen uniformly at random from the label set $\mathcal{M}$. This is a reasonable model if we assume that we choose a relatively small label set. This is a reasonable choice because the analysis shows that a significant source of randomness is contained in the "timing" information. Further, one would expect that even if we assigned unique labels to each base station, the amount of randomness does not significantly increase. This is true because, given the identity of a base station, we know its location, and for reasonable sampling intervals δ_s, the "next" base station will very likely be one of its nearest neighbors. This number of neighbors is typically small (perhaps 4) and it is this number and not the absolute number of base stations which limits the entropy rate. Given these considerations, we will stick to small label sets. As a side benefit, this requires less storage and processing.

Definition 2 (Renewal Process R, Label Process Y): Let R denote a renewal process with inter-arrival times distributed according to U. More precisely, let $F_U(x)$ denote the distribution function corresponding to U. Let $G_U(x) = \frac{1}{\mu_p}\int_{z=0}^{x}(1 - F_U(z))dz$. This new distribution is called *remaining sojourn time distribution.* Let U_1 be distributed according to G_U and let $U_2, U_3, \dots$ denote a sequence of independent random variables, distributed according to F_U. Let $R_0 = 0$ and $R_n = \sum_{i=1}^{n} U_i$, $n \geq 1$. Let $Y[i]$ denote our analytical model for $X[i]$. We pick $Y[R_n],\ n \geq 0$, uniformly at random from $\mathcal{M}$. For $i \in [R_n + 1, \dots, R_{n+1} - 1]$, we define $Y[i] = Y[R_n]$. Finally, define Y as $Y \triangleq [Y[0],\ Y[1], \dots Y[n-1]]$. We call Y the *trace*.

The reason we selected the distribution of U_1 in this particular way (and different from the distribution of all other U_i) is that this choice makes the process stationary. In particular, for $i \geq 0$ the process behaves as if the renewal process had started in the infinite past. This simplifies our analysis.

Definition 3 (Adversary Process): Consider a specific label $\mathfrak{m}_i \in \mathcal{M}$. The stochastic process of the strong adversary is defined as $A \triangleq [A[0],\ A[1], \dots A[n-1]]$, with

$$A[j] = \begin{cases} 0, & \text{if } Y[j] \neq \mathfrak{m}_i, \\ 1, & \text{if } Y[j] = \mathfrak{m}_i. \end{cases}$$

That is, the adversary is present at base stations having label $\mathfrak{m}_i$ and thus in a fraction $\frac{1}{m}$ of all base stations. The process A is stationary because it is fully determined by the stationary process Y.

B. Entropy and Entropy Rate of Y

Lemma 1: Let Y be the label process as given in Def. 2. For $n \geq 1$, $\lim_{\delta_s \to 0} \mathbb{H}(Y) = -\sum_{i=1}^{m} \frac{1}{m} \log_2 m = \log_2 m$ and $\lim_{\delta_s \to \infty} \mathbb{H}(Y) = -\sum_{i=1}^{m^n} \frac{1}{m^n} \log_2 \frac{1}{m^n} = n \log_2 m$. For $n = 1$, $\mathbb{H}(Y) = \log_2 m$. For $n = 2$, $\mathbb{H}(Y) = \log_2(m) - \frac{(\mu_p - 1)m + 1}{\mu_p m} \log_2(\frac{(\mu_p - 1)m + 1}{\mu_p m}) - \frac{m-1}{\mu_p m} \log_2(\frac{1}{\mu_p m})$.

Finally, the entropy rate $\mathcal{H}(Y)$ is given by

$$\mathcal{H}(Y) = \frac{\mathbb{H}(\{q_i\}) + \log_2(m-1)}{\mu_q},$$

where q and μ_q are defined as follows. Define $p(x) = \sum_{i=1}^{\infty} p_i x^i$. Let $q(x) = p(x)\frac{m-1}{m - p(x)}$. Let $\{q_i\}$ denote the corresponding probabilities, i.e., develop $q(x)$ as a Taylor series around $x = 0$. Let $\mathbb{H}(\{q_i\})$ be the entropy of $\{q_i\}$ and let $\mu_q = \sum i q_i$.

Due to space constraints, we refer to [1] for the proof and continue here with the discussion of the result.

Discussion. We have only given entropy expressions for $n = 1$ and 2. Although it is possible to derive expressions for larger n as well, the operationally most significant quantity is the entropy rate $\mathcal{H}(Y)$. We next discuss how $\mathcal{H}(Y)$ behaves.

For both small or large values of δ_s we can easily get accurate simplified expressions for $\mathcal{H}(Y)$. In case of not too large values of δ_s, $\mathcal{H}(Y)$ is dominated by the randomness inherent in the "timing", i.e., by the term $\mathbb{H}(\{q_i\})$. As detailed in [1], it follows that the entropy rate (per second) has in this regime the expression $a - b\log_2(\delta_s)$ (e.g., for the grid map scenario, that we will introduce in Sec. V, we have $a \approx 0.12$ and $b \approx 1/70$). This expression implies that the entropy rate goes to infinity as δ_s goes to 0. But of course, small δ_s comes at the cost of a large overhead; moreover, in every system the

978-1-4244-9848-2/11$26.00©2011 IEEE

value of δ_s is lower bounded by the inherent timing accuracy we can achieve. For very large values of δ_s the entropy is eventually dominated by the term $\log(m)$. Further, for large values of δ_s, μ_q converges to 1. Therefore, in this regime the secrecy rate scales like $\log(m)/\delta_s$.

The adversarial model is discussed in the extended version [1].

C. Relationship Between Sojourn Time Distributions

We show how to relate the sojourn time distributions of different sampling rates. This allows us to compute the sojourn time distribution once and for all and then to choose the sampling rate at a later point.

We assume that we have a distribution of sojourn times as described in [1]. For simplicity, we further assume that the basic underlying process $X(t)$ is discrete. This means that all switches take place only at multiples of a basic time unit; call it δ and $t \in \kappa\delta, \kappa \in \mathbb{N}$, e.g. δ could be a microsecond.

We now make the following distinction. There is the actual "physical" sojourn time distribution, our reference distribution, based on this time unit δ. For an actual system, we might decide for reasons of complexity (including operational system constraints) to only sample the process every δ_s time units, i.e., $\delta_s = \kappa\delta$ for some $\kappa \in \mathbb{N}$. We call this new process the sampled process X_{δ_s} and its related distribution, the sampled distribution. We are interested in the relationship between these two distributions.

We now define the procedure of sampling the sojourn time distribution more precisely.

Definition 4 (Sampling of Sojourn Time Distribution): Choose a reference process X_δ. We now fix the new sampling interval $\delta_s = \kappa\delta$. In this case, the sampled process X_{δ_s} only takes values for $t \in \mathbb{N}\delta_s$. Let i be the index of the sample and B_i the base station, the user was connected to at time $t = i\delta_s$. We collect B_i if $t \in \{(i-1)\delta_s, \dots, i\delta_s - 1\}$. We call the sojourn time the time between the first and the last sample of the same base station. With *sampling the sojourn time distribution*, we denote the step from the sojourn time distribution estimated with $\delta_s = \delta$ to the related sojourn time distribution estimated with $\delta_s = \kappa\delta$.

In order to simplify discussions, we first define our convention for notation of the distribution functions: We use small letters for the probability distribution function: $f(x) = Pr[X = x]$ with $\mu_f = \sum_{x=0}^{\infty} x f(x)$ the mean of the distribution f. For the cumulative distribution functions, we use capital letters: $F(x) = \sum_{z=-\infty}^{x} f(z)$. In the following, $x \in \{0, 1, \dots\}$.

Lemma 2 (Analytical Sampling): Let $G_{\delta_s}(x)$ be the cumulative function for the sampled remaining sojourn times. $G_{\delta_s}(x)$ is then given by $G_{\delta_s}(x) = G_\delta(\delta_n(x+1)-1), \delta_n = \frac{\delta_s}{\delta}$.

The sampled distribution of inter-arrival times is given by $F_{\delta_s}(x) = (x+1) - \frac{1}{G_{\delta_s}(0)} G_{\delta_s}(x) - \sum_{z=0}^{x-1} F_{\delta_s}(z)$, with $F_{\delta_s}(0) = 0$.

Proof: $G_{\delta_s}(x)$ is the cumulative distribution of remaining sojourn times. $g_{\delta_s}(x)$ is the probability that if one chooses one sample at random, there are still $x\delta_n$ time steps of length δ until the next switch of base station. This means that for the following $x\delta_n$ steps, the user will stay in the same base station and he will have changed before $(x+1)\delta_n$ steps. The event $g_{\delta_s}(x)$ thus includes all events from time step $x\delta_n$ until time step $(x+1)\delta_n - 1$ and thus $g_{\delta_s}(x) = \sum_{z=x\delta_n}^{(x+1)\delta_n - 1} g_\delta(z)$. We can write $G_{\delta_s}(x)$ as $G_{\delta_s}(x) = \sum_{z=0}^{x} g_{\delta_s}(z)$. Using this relation and the definition for the cumulative distribution function, we obtain that the cumulative distribution function of the remaining sojourn times equals to $G_{\delta_s}(x) = \sum_{z=0}^{(x+1)\delta_n - 1} g_\delta(z) = G_\delta((x+1)\delta_n - 1)$, which is the first part of Lem. 2. The second part is simply obtained by inverting the relation given in Def. 2. ■

This relation now allows to transforme the distributions from one sampling interval to another, thus one can easily evaluate the performance for a range of sampling intervals.

However, at this step, one still needs to simulate the traces and compute the sojourn time distribution at least once. We now aim to omit the last nontrivial simulation step and go towards modeling as presented in Sec. V-C.

Scenario	grid-net map	real map
Dimensions	$6 * 6km$	$7 * 8km$
Distance between streets	$0.3km$	different
Number of base stations	100	484
Max. speed	$17m/s$	$34m/s$
Number of cars per simulation	$480 * 30$	$840 * 30$
Entry rate	$20 * 4$cars/20sec	1400cars/500sec

Table I
SIMULATION PARAMETERS

V. EVALUATION

A. Methodology

Testing environment: We test the performance of our protocol considering users in moving cars. We produce data with the SUMO traffic simulator [6], which generates routes that mimic natural user behavior via an algorithm called dynamic user assignment[2]. In essence, this algorithm finds the quickest route instead of the shortest path. We consider two types of topology: the *grid map scenario*, which uses an artificial grid net, and the *real map scenario*, which uses a real map of Lausanne of equivalent dimensions to the grid map.

The simulated area is covered by square cells of the same size, each cell served by a base station. Each simulated base station is randomly assigned one of four different labels; thus, each label is assigned to roughly 25% of the base stations. Cars enter and exit this area at a randomly chosen border point. The precise parameters are listed in Table I.

Adversary: We considered a strong adversary who can overhear *all* parts of the trace corresponding to a specific label[3].

[2]It reflects the idea that traveling involves some time, cost or disutility that users would prefer to avoid.

[3]We also considered a weak adversary that only knows the positions in the trace where the user is connected to one specific base station, but as we found almost no performance penalty we do not report these results.

978-1-4244-9848-2/11$26.00©2011 IEEE

Theoretical model and analysis (Sec. IV) validation: We empirically calculate the distribution of the sojourn times using our simulations, and use this distribution as input for our theoretical model. We then compare the theoretically derived performance from the model with that of the simulated system.

Entropy computation and a practical challenge: For a fixed trace length n the entropy can be computed based on simulations in the following way. Assume we have generated a collection of vectors X from traces of all users in the system. This in turn allows us to estimate the probability distribution on the set of possible outcome vectors. We can then compute the entropy associated to this probability distribution. In practice, this approach quickly reaches its limits. We need to calculate the distribution on the set $\mathcal{M}^n$, which has cardinality m^n. This quickly becomes close to infeasible or at least impractical as m and n increase, and limits the scenarios we can examine through simulations. This is where the theoretical modeling can help.

B. Simulation Results

Fig. $2(a)$ and (c) show the performance of our protocol as a function of the sampling interval δ_s for the case of the grid map and the real map scenario, respectively. We plot both the results obtained through simulations (dotted curves) as well as the results obtained from the theoretical model (continuous curves). Our simulations provide results for trace lengths $n = 2, n = 3, n = 4, n = 6$ and $n = 8$; we see that these results fit quite well to the curves derived from the theoretical model. We thus proceed to use the theoretical model to derive the upper and lower bound curves also depicted in the figure, which correspond to vector lengths $n = 1$ and $n \to \infty$, respectively. The latter is derived using the entropy rate, as we discussed earlier.

As the trace length, n, grows, the curves converge towards the entropy rate (which, as discussed earlier, gives a lower bound for any length): we closely approach this curve even for relatively short lengths, e.g., $n = 8$. Curves for smaller n seem to deliver higher secrecy rates. However, this is misleading: these computations only apply if we collect a set of such vectors well-separated in time, but they are meaningless if we sample consecutive vectors, due to the dependence of the values such vectors take. Thus, the entropy rate gives a more realistic estimate of the expected performance. E.g., in the grid map scenario, we expect to collect 128 random bits in approx. 25 minutes, if we sample three times per minute.

Fig. $2(b)$ shows how the presence of an adversary affects performance: it reduces the secrecy rate by approx. 40%. Indeed, by overhearing all communications of 25% of the base stations, the adversary can deduce a lot more. As an extreme case, consider the example of $m = 2$: an adversary eavesdropping half the system area can in fact reconstruct the whole trace.

C. Mobility Model for Sojourn Time Distribution

We use a mobility model introduced by [7], that provides sojourn time distributions for cellular systems in the following way: We fix the size and the shape of a cell which is covered by straight street segments of different length. The speed along one segment is supposed to be constant and after each step, the car has the possibility to change direction and speed (while an average speed is maintained). The length, the speed and the change of direction are drawn independently from their corresponding distributions. The time spend by the car inside the cell is the sojourn time.

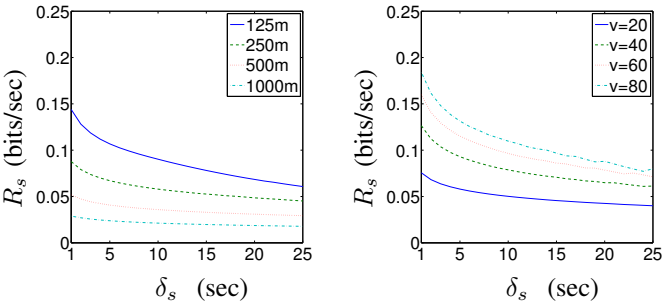

Figure 3. Entropy rate for (a) average speed of $10\frac{km}{h}$, base station diameter of 125, 250, 500 and $1000m$, (b) base station diameter of $500m$, average speed of 20, 40,60 and $80\frac{km}{h}$.

The mobility model in [7] allows us to obtain the sojourn time distribution for various scenarios in an easy way. We apply the transformation in Sec. IV on this distribution in order to obtain the sojourn time distribution for a range of sampling intervals. These distributions can then be used in order to compute the entropy rate for various setups, as described in [1]. As mentioned earlier, the entropy rate is a realistic estimation of the ultimate performance for our scheme.

In Fig. 3, we show, how the entropy rate behaves for different scenarios. One can observe that the entropy rate scales with the cell size. The scaling with the average speed is a little less straight forward, as the variance of the speed distribution plays an important role.

VI. Related Work

Limitations of traditional key management based on public key cryptography spurred work towards alternative symmetric key establishment methods. The motivation has been to avoid trusted third parties and public key cryptography (e.g., the Diffie-Hellman algorithm [8]). Information-theoretic schemes for key generation based on correlated information have been the basis: simply put, two legitimate parties observe a source the adversary cannot, even if it can intercept messages they exchange [9]. The adversary can be a passive eavesdropper (as is also the case for the DH protocol); the possibility of key establishment in the presence of active adversaries is proven [10].

Many schemes were proposed, leveraging the wireless channel properties to establish common information the eavesdropper cannot obtain, such as [11], [12], [13], [14], [15], [16], [17], [18], [19].

Physical layer based schemes may require special hardware or modification of the wireless transceivers. Even though they

978-1-4244-9848-2/11$26.00©2011 IEEE

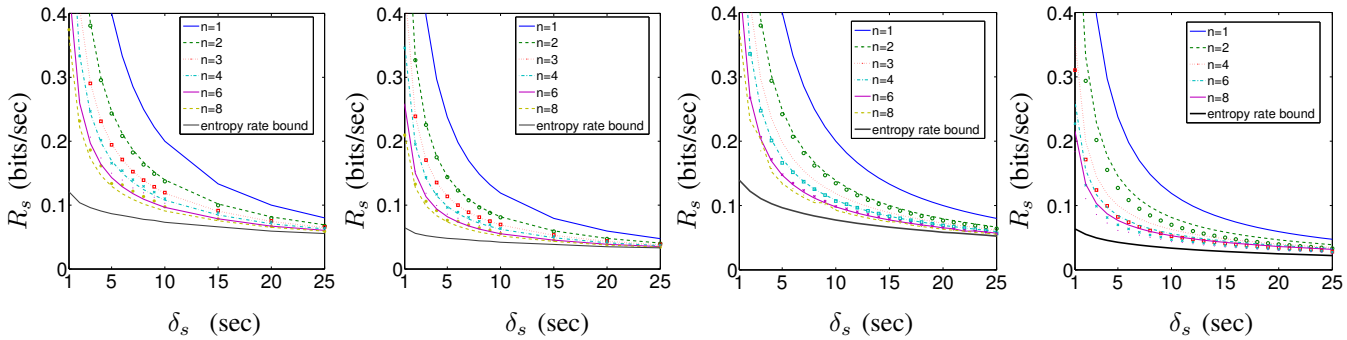

Figure 2. Grid map (a) without adversary and (b) with adversary, real map scenario (c) without adversary and (d) with adversary.

can offer significant secrecy rates (e.g., 10 bits/s by [20]), they are limited in local pair-wise operation Without any transceiver modification, two devices could extract common information by tracking "one time frames", i.e., wireless transmissions received at the first attempt [21]; an adversary would need to correctly eavesdrop all wireless transmissions for an extended time period. But none of the above allows two remote devices that are not connected across the wireless medium to establish a shared key. This would be the case for a broad amount of mobile applications, beyond that what ad hoc 802.11, WSN, or UWB can support. We close this gap: our scheme is independent of the wireless communication specifics, leveraging node mobility and traces of connectivity with wireless infrastructure; and it is deployable in existing systems with no transceiver modification. For references for mobility modeling, we refer to [7] and references herein.

VII. Conclusions and Discussion

Our base position in this paper is that mobility has inherent randomness that can be exploited to establish common random bits at low cost. We proposed a protocol that leverages mobility for secret key generation, and can be deployed in current cellular systems with minimal modifications of existing protocols.

Our scheme turns out to be promising: even though it does not yield high secrecy rates, it operates at low cost. Indeed, the 15-25 minutes, in the mobility scenarios we investigated, might at first seem long for the establishment of 128 bit secret keys. But this is well below the average daily commute time for many users, and the overall operation requires $3-6$ samples per minute, a low overhead for current mobile devices but also easy to handle for wireless infrastructures. These almost "free" random bits can be used to enhance and complement other security systems, when needed; even in the presence of a strong adversary.

Finally, mobility can be an incessant source of randomness, and thus secrecy: users can get essentially "for free" a key established while on the move. Further, they can have a continuous accumulation of secret bits over time, and refresh older (and perhaps compromised) keys, along the lines of the idea to recover loss of secrecy with newly dynamically established secrets in [21].

References

[1] C. Neuberg, P. Papadimitratos, C. Fragouli, and R. Urbanke, "A mobile world of security," 2010. [Online]. Available: http://infoscience.epfl.ch/record/150219

[2] T. Cover and J. Thomas, *Elements of Information Theory*. Wiley, 2006.

[3] C. Bennett, G. Brassard, and U. Maurer, "Generalized privacy amplification," *IEEE Trans. on Inf. Th.*, 1995.

[4] J. Carter and M. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, 1997.

[5] S. Halevi and H. Krawczyk, "Strengthening digital signatures via randomized hashing," in *Advances in Cryptology, LNCS 4117*. Springer, 2005, pp. 41–59.

[6] D. Krajzewicz, M. Bonert, and P. Wagner, "The open source traffic simulation package - SUMO," 2006.

[7] P. Bratanov and E. Bonek, "Mobility model of vehicle-borne terminals in urban cellular systems," *IEEE Trans. Veh. Technol.*, Jul. 2007.

[8] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. on Inf. Th.*, Nov. 1976.

[9] U. Maurer, "Secret key agreement by public discussion from common information," *IEEE Trans. on Inf. Th.*, May 1993.

[10] U. Maurer and S. Wolf, "Secret-key agreement over unauthenticated public channels: Definitions and a completeness result," *IEEE Trans. on Inf. Th.*, Apr. 2003.

[11] J. Hershey, A. Hassan, and R. Yarlagadda, "Unconventional cryptographic keying variable management," *IEEE Trans. on Communications*, Jan. 1995.

[12] M. Tope and J. McEachen, "Unconditionally secure communications over fading channels," in *IEEE MILCOM*, 2001.

[13] B. Azimi-Sadjadi, A. Kiayias, and A. Mercado, "Robust key generation from signal envelopes in wireless networks," in *ACM CCS*, Alexandria, Virginia, USA, 2007.

[14] S. Mathur, W. Trappe, and N. Mandayam, "Radio-telepathy: Extracting a secret key from an unauthenticated wireless channel," in *ACM MobiCom*, Sept. 2008.

[15] J. Croft, N. Patwari, and S. Kasera, "Robust uncorrelated bit extraction methodologies for wireless sensors," in *ACM IPSN*, Stockholm, Sweden, Apr. 2010.

[16] R. Wilson, D. Tse, and R. Scholtz, "Channel identification: Secret sharing using reciprocity in ultrawideband channels," *IEEE Trans. on Information Forensics and Security*, Sept. 2007.

[17] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure information transmission for mobile radio," *IEEE Comm. Letters, IEEE*, Feb. 2000.

[18] A. Sayeed and A. Perrig, "Secure wireless communications: Secret keys through multipath," in *IEEE ICASSP*, Las Vegas, NV, USA, Mar. 2008.

[19] T. Aono, K. Higuchi, T. Ohira, B. Komiyama, and H. Sasaoka, "Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels," *IEEE Trans. on Antennas and Propagation*, Nov. 2005.

[20] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Trans. Information Forensics and Security*, Jun. 2010.

[21] S. Xiao, W. Gong, and D. Towsley, "Secure wireless communication with dynamic secrets," *IEEE INFOCOM*, 2010.

978-1-4244-9848-2/11$26.00©2011 IEEE